

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-056955

(43)Date of publication of application : 25.02.2000

(51)Int.Cl. G06F 9/06

(21)Application number : 10-219999

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 04.08.1998

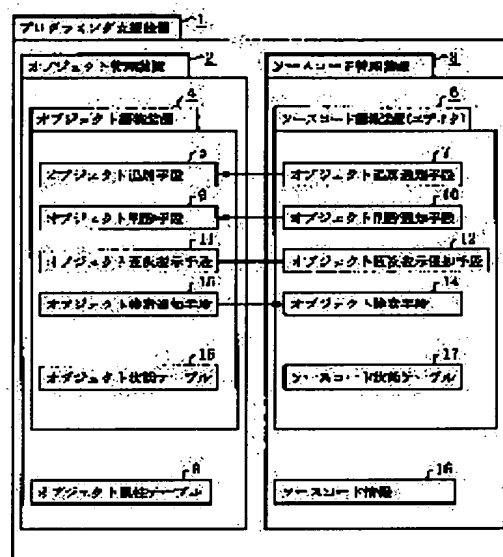
(72)Inventor : KAYANO SHIGEMI

(54) OBJECT MANAGING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an object managing method for clearly developing a program by arranging objects such as variable defined by a programmer defines in programming, thereby displaying them.

SOLUTION: An object managing device 2 having an object editing device 4 and an object attribute table inside, and a source code managing device 3 having a source code editing device (editor) 6 and source code information inside are provided. The objects such as variable peculiar to a program which is defined by a programmer are unitarily managed by cooperating two devices on a real time.



LEGAL STATUS

[Date of request for examination] 13.07.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開2000-56955

(P2000-56955A)

(43)公開日 平成12年2月25日(2000.2.25)

(51)Int. Cl.

G06F 9/06

識別記号

530

FI

G06F 9/06

テマコード(参考)

530P 5B076

審査請求 有 請求項の数12 OL (全14頁)

(21)出願番号 特願平10-219999

(22)出願日 平成10年8月4日(1998.8.4)

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 荻野 重実

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(74)代理人 100102439

弁理士 宮田 金雄 (外2名)

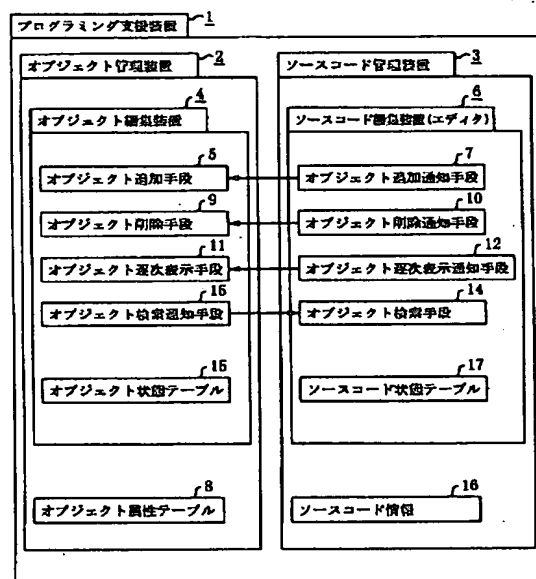
Fターム(参考) 5B076 DB05 DC04 EC08

(54)【発明の名称】 オブジェクト管理方法

(57)【要約】

【課題】 プログラミングにおいて、プログラマーが定義した変数等のオブジェクトを整理した形式で表示することによって明瞭なプログラム開発を行なうためのオブジェクト管理方法を得る

【解決手段】 オブジェクト編集装置、オブジェクト属性テーブルを内部的に持つオブジェクト管理装置と、ソースコード編集装置(エディタ)、ソースコード情報を内部的に持つソースコード管理装置を備え、両者がリアルタイムに連携することにより、プログラマーが定義したプログラム固有の変数等のオブジェクトを一元管理する。



1

【特許請求の範囲】

【請求項1】 ソースコードを入力するステップ、上記ソースコードに対応するオブジェクトIDを自動生成するステップ、上記ソースコードが定義文内にあれば定義日付を設定し参照文内にあれば参照日付を設定するステップ、区切りコードを入力することにより上記ソースコードがオブジェクトとして確定するステップ、確定したオブジェクトの属性情報をオブジェクト属性テーブルに格納しオブジェクト一覧として表示するステップからなることを特徴とするオブジェクト管理方法。

【請求項2】 ソースコードを入力し区切りコードを入力することにより上記ソースコードがオブジェクトとして確定するステップ、上記オブジェクトに対応するオブジェクトIDを自動生成するステップ、上記オブジェクトが定義文内にあれば定義日付を設定し参照文内にあれば参照日付を設定するステップ、オブジェクトの属性情報をオブジェクト属性テーブルに格納しオブジェクト一覧として表示するステップからなることを特徴とするオブジェクト管理方法。

【請求項3】 上記オブジェクトが定義文内にあるとき 20 対応する定義日付が上記オブジェクト属性テーブルに既に存在するときはアラームを発することを特徴とする請求項1又は請求項2記載のオブジェクト管理方法。

【請求項4】 上記オブジェクトが参照文内にあるとき 対応する定義日付が上記オブジェクト属性テーブルにないときはアラームを発することを特徴とする請求項1～請求項3のいずれかに記載のオブジェクト管理方法。

【請求項5】 上記アラームは上記入力したオブジェクトが反転表示されることを特徴とする請求項3又は請求項4記載のオブジェクト管理方法。

【請求項6】 上記アラームは上記入力したオブジェクトがブリンク表示されることを特徴とする請求項3又は請求項4記載のオブジェクト管理方法。

【請求項7】 画面に表示された上記ソースコード上の上記オブジェクトにカーソルを位置づけるとそのオブジェクトがオブジェクト一覧として表示されることを特徴とする請求項1～請求項6のいずれかに記載のオブジェクト管理方法。

【請求項8】 画面に表示された上記ソースコード上の 40 上記オブジェクトにカーソルを位置づけそのオブジェクトの名前を変更したときは上記オブジェクト属性テーブル上の属性データも変更されることを特徴とする請求項1～請求項7のいずれかに記載のオブジェクト管理方法。

【請求項9】 上記オブジェクト一覧のオブジェクトの定義日付にカーソルを位置づけると画面に表示された上記ソースコード上の定義文内にある上記オブジェクトが反転表示されることを特徴とする請求項1～請求項8のいずれかに記載のオブジェクト管理方法。

【請求項10】 上記オブジェクト一覧のオブジェクト 50

2

の参照日付にカーソルを位置づけると画面に表示された上記ソースコード上の参照文内にある上記オブジェクトが反転表示されることを特徴とする請求項1～請求項8のいずれかに記載のオブジェクト管理方法。

【請求項11】 上記オブジェクト一覧のオブジェクトの名称にカーソルを位置づけると画面に表示された上記ソースコード上の定義文内にある上記オブジェクトと参照文内にある上記オブジェクトとがそれぞれ上記ソースコード上の他のオブジェクトとは識別可能に表示されることを特徴とする請求項1～請求項8のいずれかに記載のオブジェクト管理方法。

【請求項12】 上記ソースコード上の先頭に近いオブジェクトから順次オブジェクトが画面表示されることを特徴とする請求項10又は請求項11に記載のオブジェクト管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、ソフトウェア開発を支援するために、プログラム中のオブジェクトをリアルタイムに一元管理し、オブジェクトの一覧表示、属性確認、定義・参照確認等によって効率的なプログラム開発を行なうためのオブジェクト管理方法に関するものである。

【0002】

【従来の技術】一般に、変数名、I/O部品、関数、ロケーション等のオブジェクトを定義あるいは参照してロジックを組み立てるプログラミング作業において、各オブジェクトがどこでどのように定義されどこで参照されているかを把握するのは各プログラマーの記憶に頼るところが多い。またプログラマーが記憶していないものや記憶が曖昧なものに関してはその都度プログラマーがその定義箇所、参照箇所を検索するなどして確認しなければならない。この作業はプログラミング規模が大きければ大きいほど煩雑なものになる。

【0003】特開平5-313878号公報に記載された発明は、上記のようなソフトウェア開発を支援するためのものであり、変数名をテーブルに登録することによってコーディング誤りを逐次チェックし、プログラマーのコーディング作業を支援している。

【0004】

【発明が解決しようとする課題】従来のオブジェクト管理方法は、チェックするオブジェクトが変数、予約語、区切り文字に限られており、I/O部品、関数、ロケーションといったオブジェクトに関しては支援していない。またテーブルへの登録はその変数が新たに定義された場合のみであり、変更、削除された場合については支援していない。さらに変数のテーブルを一覧表示する手段がなくプログラマーがすべての変数のテーブルをイメージするのが難しいという問題点があった。

【0005】この発明は、上記のような問題点を解消す

3

るためになされたもので、変数に限らずプログラムで使用するすべてのオブジェクトに関して定義および参照がコード上で作成、変更、削除された際に、対応する情報をオブジェクト属性テーブルとして格納、一元管理し、そのオブジェクトの属性を逐次一覧表示したり定義箇所、参照箇所を検索することによってプログラマーを支援することを目的とする。

【0006】

【課題を解決するための手段】この発明に係るオブジェクト管理方法は、ソースコードを入力するステップ、上記ソースコードに対応するオブジェクトIDを自動生成するステップ、上記ソースコードが定義文内にあれば定義日付を設定し参照文内にあれば参照日付を設定するステップ、区切りコードを入力することにより上記ソースコードがオブジェクトとして確定するステップ、確定したオブジェクトの属性情報をオブジェクト属性テーブルに格納しオブジェクト一覧として表示するステップからなるものである。

【0007】また、ソースコードを入力し区切りコードを入力することにより上記ソースコードがオブジェクトとして確定するステップ、上記オブジェクトに対応するオブジェクトIDを自動生成するステップ、上記オブジェクトが定義文内にあれば定義日付を設定し参照文内にあれば参照日付を設定するステップ、オブジェクトの属性情報をオブジェクト属性テーブルに格納しオブジェクト一覧として表示するステップからなるものである。

【0008】さらに、上記オブジェクトが定義文内にあるとき対応する定義日付が上記オブジェクト属性テーブルに既に存在するときはアラームを発するものである。

【0009】また、上記オブジェクトが参照文内にあるとき対応する定義日付が上記オブジェクト属性テーブルにないときはアラームを発するものである。

【0010】さらにまた、上記アラームは上記入力したオブジェクトが反転表示されるものである。

【0011】また、上記アラームは上記入力したオブジェクトがブリンク表示されるものである。

【0012】さらに、画面に表示された上記ソースコード上の上記オブジェクトにカーソルを位置づけるとそのオブジェクトがオブジェクト一覧として表示されるものである。

【0013】また、画面に表示された上記ソースコード上の上記オブジェクトにカーソルを位置づけそのオブジェクトの名前を変更したときは上記オブジェクト属性テーブル上の属性データも変更されるものである。

【0014】さらにまた、上記オブジェクト一覧のオブジェクトの定義日付にカーソルを位置づけると画面に表示された上記ソースコード上の定義文内にある上記オブジェクトが反転表示されるものである。

【0015】また、上記オブジェクト一覧のオブジェクトの参照日付にカーソルを位置づけると画面に表示され

4

た上記ソースコード上の参照文内にある上記オブジェクトが反転表示されるものである。

【0016】さらに、上記オブジェクト一覧のオブジェクトの名称にカーソルを位置づけると画面に表示された上記ソースコード上の定義文内にある上記オブジェクトと参照文内にある上記オブジェクトとがそれぞれ上記ソースコード上の他のオブジェクトとは識別可能に表示されるものである。

【0017】また、上記ソースコード上の先頭に近いオブジェクトから順次オブジェクトが画面表示されるものである。

【0018】

【発明の実施の形態】実施の形態1. 図1はこの発明の一実施の形態としてのオブジェクト管理方法を実施するためのプログラミング支援装置全体の構成図を示す。1はプログラミング支援装置、2はオブジェクトを一元管理するオブジェクト管理装置、3はオブジェクト管理装置2と逐次連携してソースコードを管理するソースコード管理装置である。

【0019】オブジェクト管理装置2において、4はオブジェクトの表示等の処理を行なうオブジェクト編集装置、5はソースコード編集装置（エディタ）6のオブジェクト追加通知手段7から通知されたオブジェクトをオブジェクト属性テーブル8に追加するオブジェクト追加手段、9はソースコード編集装置（エディタ）6のオブジェクト削除通知手段10から通知されたオブジェクトをオブジェクト属性テーブル8から削除するオブジェクト削除手段、11はソースコード編集装置（エディタ）6のオブジェクト逐次表示通知手段12から通知されたオブジェクトをオブジェクト属性テーブル8から検索してオブジェクト編集装置4に表示するオブジェクト逐次表示手段、13はオブジェクトの定義箇所、参照箇所をソースコード編集装置（エディタ）6のオブジェクト検索手段14に通知するオブジェクト検索通知手段である。

【0020】15はオブジェクト編集装置4のある時点での状態を格納するオブジェクト状態テーブル、8はすべてのオブジェクトの属性に関する一元管理情報であるオブジェクト属性テーブルである。またソースコード管理装置3において、6はソースコードの編集その他の処理を行なうソースコード編集装置（エディタ）、7はソースコード編集装置（エディタ）6によって追加されたオブジェクトをオブジェクト編集装置4のオブジェクト追加手段5に通知するオブジェクト追加通知手段、10はソースコード編集装置（エディタ）6によって削除されたオブジェクトをオブジェクト編集装置4のオブジェクト削除手段9に通知するオブジェクト削除通知手段である。

【0021】12はソースコード編集装置（エディタ）6によってカーソルが位置づけられたオブジェクトをオ

5

プロジェクト編集装置4のオブジェクト逐次表示手段11に通知するオブジェクト逐次表示通知手段、14はオブジェクト編集装置4のオブジェクト検索通知手段13から通知されたオブジェクトをソースコード情報16の中から検索してソースコード編集装置(エディタ)6に表示するオブジェクト検索手段、17はある時点でのソースコード編集装置(エディタ)6の状態を格納したソースコード状態テーブル、16はすべてのソースコードに関する情報であるソースコード情報である。

【0022】図2にオブジェクト編集装置4による編集結果の一例として変数の場合のオブジェクト一覧としての表示画面を示す。変数以外のオブジェクトには、IO部品、関数、ロケーションがある。各オブジェクトの表示画面の切替はGUIを用いて簡単に行なうことができる。またオブジェクトの属性をキーとして表示をソートすることができる。また表示画面の一覧表におけるカラムのうち入力できるのはコメントだけであり、他のカラムは表示のみ可能である。

【0023】図3にオブジェクト属性テーブルの情報形式を示す。図中の定義ID/日付、参照ID/日付とは、後述するソースコード情報におけるオブジェクトIDとその生成日付が入る。また参照ID/日付は可変個存在する。

【0024】図4はオブジェクト編集装置4のある時点での状態を示すオブジェクト状態テーブルの構成である。図中のオブジェクト名バッファ、定義数、参照数はそれぞれ、その時点で選択されているオブジェクトに関する名前、定義ID/日付の数(0または1)、参照ID/日付の数(0以上)が入る。また参照検索カウントは選択されているオブジェクトの参照箇所検索がその時点で何番目の参照IDであるかを示すものである。これは参照箇所検索コマンドを行なうごとに1ずつインクリメントされ、最後の参照IDを指している場合は1に戻る。また選択されているオブジェクトが変更された場合も1に戻る。

【0025】図5にソースコード編集装置(エディタ)により編集するための入力画面を示す。画面に表示されたソースコードに追加修正することにより編集することができる。図5ではソースコードが2つ表示されているが、2画面に限定されるものではない。

【0026】図6にソースコード情報を示す。上部のソース1の表示画面に表示された2行のソースコードがソースコード編集装置(エディタ)6によってソースコードを表示した画面、下部の枠内がテキスト形式で表した実際のソースコード情報である。下部枠内のリストからわかるように、ソースコード編集装置(エディタ)6は実際のソースコード情報のうち奇数行目のみを上部の表示画面に表示する。ソースコード編集装置(エディタ)6では表示されない偶数行目には付加情報としてオブジェクトID(ソースコード上の各オブジェクト記述に対

6

して1対1に対応する一意なID番号)が入っている。このオブジェクトIDには、プログラム上固有のオブジェクト以外のもの(予約語など)には0が入り、固有のオブジェクトとは区別をつけている。例えば、図6のintは0Lで表されている。

【0027】また図7にソースコード編集装置(エディタ)6の状態であるソースコード状態テーブルを示す。新、旧、それぞれ、予約語バッファが1つ、オブジェクトID配列(オブジェクトID1、オブジェクトID2、...)が可変個、オブジェクトIDが1個存在する。ここで予約語とは、変数や関数を定義する言語特有の文字列を意味する。これらの内容をその時点でのソースコード編集装置(エディタ)6の状態別に説明する。

【0028】まずカーソルが予約語に位置づけられている場合、予約語バッファにその予約語が入り、予約語に続くオブジェクトが1つ以上ある場合にはそのオブジェクトIDがオブジェクトID配列(オブジェクトID1、オブジェクトID2、...)に入り、オブジェクトIDには0が入る。予約語に続くオブジェクトが1つもない場合にはオブジェクトID配列には0が入る。

【0029】次にカーソルがオブジェクトに位置づけられている場合、予約語バッファには空白、オブジェクトID配列には0が入り、オブジェクトIDにそのオブジェクトのIDが入る。またカーソルが予約語にもオブジェクトにも位置づけられていない場合は、予約語バッファには空白、オブジェクトID配列、オブジェクトIDにはいずれも0が入る。

【0030】ここで、新、旧、2つの領域には、ソースコード状態(以後、ソースコード状態とは、ソースコード自体とカーソル位置の両者を意味する)が変更されるたびに、変更前のソースコード状態が旧、変更後のソースコード状態が新、に格納されることになる。またオブジェクトIDカウントには、その時点での全オブジェクトIDの最大値が入っており、新規にオブジェクトIDを追加する際に参照される。

【0031】本プログラミング支援装置の処理の流れを示す。処理としては、ソースコード編集装置(エディタ)6主導のものと、オブジェクト編集装置4主導のものとがある。まず、ソースコード編集装置(エディタ)6主導の処理概要を図8に示す。図のステップ801はソースコード状態が変更されるたびごとにこれらの処理が行なわれることを示している。また複数文字をまとめて追加、あるいは削除した場合はその文字数分の処理が行なわれることになる。

【0032】最初の処理は、ソースコード状態テーブルの格納である(ステップ802)。これはその時点のカーソル位置とその前後のソースコードからソースコード状態テーブルの新予約語バッファ、新オブジェクトID配列、新オブジェクトID、に変更後の各情報を格納する。その後、ソースコード状態変化がソースコード自体

7

の変更である場合のみ以下のステップ804～813の処理が行なわれる。

【0033】まず、旧予約語バッファに何か入っている場合(ステップ804)、つまりカーソルが予約語に位置づけられている状態からその予約語文字列を変更した場合であり、予約語が予約語でなくなってしまう状態、それに続くオブジェクトがあれば(ステップ805)、その定義を削除する処理を行なう(ステップ806)。逆に旧予約語バッファに何も入っておらず(ステップ804)、また旧オブジェクトIDに何か入っている場合(ステップ807)、つまりカーソルがオブジェクトに位置づけられている状態からそのオブジェクト名を変更した状態、そのオブジェクトの定義または参照を削除する処理を行なう(ステップ808)。

【0034】続いて新予約語バッファに何か入っている場合(ステップ809)、つまり予約語文字列が書かれた場合、それに続くオブジェクトがあれば(ステップ810)、その定義を追加する処理を行なう(ステップ811)。逆に新予約語バッファに何も入っておらず(ステップ809)、また新オブジェクトIDに何か入っている場合(ステップ812)、つまり新しいオブジェクトが書かれた状態、そのオブジェクトの定義または参照を追加する処理を行なう(ステップ813)。引き続いて、ソースコードの変更がされた場合、およびカーソルが移動されたのみの場合、の共通の処理が行なわれる。旧オブジェクトと新オブジェクトが異なる場合(ステップ814)、つまりカーソルが位置づけられているオブジェクトが変化した場合、新オブジェクトを表示する処理を行なう(ステップ815)。そして最後に新しい状態のバッファを旧の状態のバッファにオーバーライトすることに(ステップ816)より、ソースコード状態テーブルを更新する(ステップ816)。

【0035】図8のステップ811やステップ813のオブジェクト追加処理を図9に示す。図8のステップ811の場合はすべてのオブジェクトに対してこの処理が行なわれる。図1のソースコード編集装置(エディタ)6のオブジェクト追加通知手段7、およびオブジェクト編集装置4のオブジェクト追加手段5に相当する。まずオブジェクト追加通知手段7の処理を示す。ソースコードが変更されたことにより新たにオブジェクトが追加された場合、そのオブジェクトに対応するオブジェクトIDがソースコード情報になかったら(ステップ901)、オブジェクトIDカウントを用いて対応するオブジェクトIDを作成(ステップ902)した後、オブジェクトIDカウントを1だけインクリメントする(ステップ903)。

【0036】その後、ソースコード情報中でそのオブジェクトの前後から、そのオブジェクトの種別(変数、IO部品、関数、ロケーション)、およびそのオブジェクトが定義されているのか参照されているのかを得る(ス

8

テップ904)。ここでオブジェクトの種別を得るアルゴリズムは言語によって異なるが、例えばC言語の場合、IO命令に続いていればIO部品の参照、予約語に続いて、「文字列(…){ }」とあれば関数定義、「文字列() ;」は関数参照、「文字列:」はロケーション定義、分岐命令に続く文字列はロケーション参照、予約語に続く文字列のみは変数定義、上記以外は変数参照、という具合に判定可能である。その結果、定義であった場合(ステップ905)、その定義箇所があるソースブロック内で同じオブジェクト名の参照箇所を検索した結果あった場合(ステップ906)、その参照箇所に対応する定義箇所がなければ(ステップ907)、そのオブジェクトIDを関連オブジェクトIDとし(ステップ912)、定義箇所があればステップ906に戻る。

【0037】またステップ906の検索の結果、なければ関連オブジェクトIDを0としておく(ステップ911)。またステップ904の判定の結果、参照であった場合は(ステップ905)、その参照箇所があるソースブロック内で同じオブジェクト名の定義箇所を検索し(ステップ908)、もしなければそのソースブロックを含むさらに大きいソースブロック内を検索していく(ステップ909、910)。ただしステップ808の検索で新たなソースブロック内に入り込んで検索することはしない。検索の結果、定義箇所があればそのオブジェクトIDを関連オブジェクトIDとし(ステップ912)、なければ関連オブジェクトIDを0としておく(ステップ911)。以上のような処理の結果得られた情報(オブジェクト名、オブジェクトID、オブジェクト種別、定義or参照フラグ、関連オブジェクトID)をオブジェクト編集装置4のオブジェクト追加手段5に通知する(ステップ913)。

【0038】オブジェクト追加手段5では、まず関連オブジェクトIDが0でなければ(ステップ914)、通知されたオブジェクト種別、定義or参照フラグ、関連オブジェクトIDによって、オブジェクト属性テーブルから関連オブジェクトを検索する(ステップ915)。また逆に関連オブジェクトIDが0の場合(ステップ914)、オブジェクト属性テーブルの通知されたオブジェクト種別にオブジェクト名を追加作成する(ステップ916)。さらに通知された定義又は参照フラグによって(ステップ917)、通知されたオブジェクトIDとその時点での日付を、オブジェクト属性テーブルの定義日付と定義ID/日付(ステップ918)、または参照日付と参照ID/日付(ステップ919)に書き込む。

【0039】次に図8のステップ806や808のオブジェクト削除処理を図10に示す。ステップ806の場合はすべてのオブジェクトに対してこの処理が行なわれる。図1のソースコード編集装置(エディタ)6のオブジェクト削除通知手段10、およびオブジェクト編集装置4のオブジェクト削除手段9に相当する。まずオブジェクト削除通知手段10の処理を示す。ソースコード情報中でオ

9

プロジェクトの前後からそのオブジェクトの種別(変数、IO部品、関数、ロケーション)、そのオブジェクトが定義されているのか参照されているのか、およびオブジェクトIDを得て(ステップ1001)、オブジェクト編集装置4のオブジェクト削除手段9に通知する(ステップ1002)。また続いてソースコードの変更によってオブジェクト文字列が完全に削除された場合は(ステップ1003)ソースコード情報から対応するオブジェクトIDを削除する(ステップ1004)。

【0040】一方、オブジェクト削除手段9では、通知されたオブジェクト種別、定義or参照フラグ、オブジェクトIDをオブジェクト属性テーブルから検索して(ステップ1005)オブジェクトIDを削除する(ステップ1006)。またそのオブジェクトに対して定義ID、参照IDが1つもなくなってしまう(ステップ1007)オブジェクト属性テーブルからオブジェクト自体を削除する(ステップ1008)。逆に定義ID、参照IDが残っている場合は、定義又は参照フラグによって(ステップ1009)定義日付又は参照日付を削除する(ステップ1010、1011)。ここでもし参照日付を削除した場合は、参照IDが1つ以上残っている場合は(ステップ1012)、参照ID/日付の中で最新のものを参照日付にコピーする(ステップ1013)。

【0041】図8のステップ815のオブジェクト逐次表示処理を図11に示す。図1のソースコード編集装置(エディタ)6のオブジェクト逐次表示通知手段12、およびオブジェクト編集装置4のオブジェクト逐次表示手段11に相当する。まずオブジェクト逐次表示通知手段12の処理を示す。これは単に新オブジェクトIDをオブジェクト編集装置4のオブジェクト逐次表示手段11に通知する(ステップ1101)。オブジェクト逐次表示手段11は、通知されたオブジェクトIDをオブジェクト属性テーブル8から見つけ出しオブジェクト編集装置4に選択表示する(ステップ1102)。またオブジェクト状態テーブル15を更新する(ステップ1103)。

【0042】次に、オブジェクト編集装置4主導の処理概要を図12に示す。図1のオブジェクト編集装置4のオブジェクト検索通知手段13、およびソースコード編集装置(エディタ)6のオブジェクト検索手段14に相当する。オブジェクト編集装置4のオブジェクト検索通知手段13において、定義箇所検索コマンドが実行された場合、選択されているオブジェクトの定義IDを取り出しオブジェクトIDとする(ステップ1201)。また参照箇所検索コマンドが実行された場合は、参照検索カウンタが指す参照IDを取り出しオブジェクトIDとした(ステップ1202)後、参照検索カウンタが参照数と同じであれば(ステップ1203)参照検索カウンタを1とし(ステップ1204)、異なれば参照検索カウンタを1だけインクリメントする(ステップ1205)。以上によって得られたオブジェクトIDをソースコード編集装置(エディ

10

タ)6のオブジェクト検索手段14に通知する(ステップ1206)。ソースコード編集装置(エディタ)6のオブジェクト検索手段14では、通知されたオブジェクトIDをソースコード情報16から検索し、その部分を選択表示する(ステップ1207)。

【0043】以上のような内部情報および内部処理によって実現される画面を示す。図13は、ソースコード編集装置(エディタ)6によって変数の定義を記述しているところである。ソースコード編集装置(エディタ)6によって、変数などのオブジェクトの定義を記述するとリアルタイムに定義情報がオブジェクト編集装置4に通知されてオブジェクト属性テーブル8に格納されると同時に、そのオブジェクトが表示される。図13では、

「nLoopCo」を例として取り上げているが、この「nLoopCo」をソースコードとして入力するとき、先頭のnを入力した時点でそのnを定義情報としてリアルタイムにオブジェクト編集装置4に通知されてオブジェクト属性テーブル8に格納されると同時に、そのオブジェクトが表示される。nLoopCoまで入力した時点において、「nLoop」は既にオブジェクト属性テーブル8に存在するため、音によるアラームを発生させるか表示画面上でnLoopを反転表示させるかブリンクさせて操作者に注意を喚起することが可能である。

「nLoop」まで入力した後カーソルを「nLoop」以外に移動させた時点で上記アラームを発生させることも可能である。

【0044】図14は、ソースコード編集装置(エディタ)6によってロケーションの参照を記述しているところである。図13と同様に、オブジェクトの参照を記述するとリアルタイムに参照情報がオブジェクト編集装置4に通知されてオブジェクト属性テーブル8に格納されると同時に、そのオブジェクトが表示される。図15は、単にカーソルを関数に位置づけた場合である。この場合もリアルタイムにその関数が表示される。最後にオブジェクト編集装置において、IO部品の参照箇所検索コマンドを実行した場合を図16に示す。この場合は選択されているIO部品が参照されている箇所をソースコード編集装置(エディタ)6の表示画面上に選択表示している。ここで、参照箇所が複数存在する場合は、参照箇所と表示されたボタンをクリックすることにより、ソースコード上の先頭に近いオブジェクトから順次当該オブジェクトが参照されている箇所をソースコード編集装置(エディタ)6の表示画面上に選択表示する。

【0045】実施の形態2。上記実施の形態1においては、ソースコード編集装置(エディタ)6によって、変数などのオブジェクトの定義を記述するとき、先頭の1文字が入力された時点でリアルタイムに定義情報がオブジェクト編集装置4に通知されてオブジェクト属性テーブル8に格納されると同時に、そのオブジェクトが表示されていたが、実施の形態2においては、ソースコード

11

の一単位としての定義情報である名前等が入力された後、例えばリターンキーを押下すること又はブランクを入力することにより、区切りコードが入力された時点で初めてリアルタイムに定義情報がオブジェクト編集装置4に通知されてオブジェクト属性テーブル8に格納されると同時に、そのオブジェクトを表示させるものである。

【0046】

【発明の効果】この発明は以上述べた様に構成されているので、以下に述べる効果を有する。

【0047】入力しているソースコードに対応するオブジェクトIDを自動生成し、ソースコードが定義文内にあれば定義日付を設定し参照文内にあれば参照日付を設定する様に構成したので、オブジェクトをリアルタイムに一元管理することができる。

【0048】また、ソースコードを入力し区切りコードを入力することにより上記ソースコードがオブジェクトとして確定し、オブジェクトに対応するオブジェクトIDを自動生成する様に構成したので、入力したオブジェクトを一元管理することができる。

【0049】さらに、オブジェクトに対応する定義日付がオブジェクト属性テーブルにないときはアラームを発する様に構成したので、オブジェクトの重複使用を避けることができる。

【0050】また、オブジェクトが参照文内にあるとき対応する定義日付がオブジェクト属性テーブルにないときはアラームを発する様に構成したので、未定義での使用を避けることができる。

【0051】さらにまた、アラーム時は入力したオブジェクトが反転表示される様に構成したので、容易にオブジェクトの重複使用又未定義での使用を避けることができる。

【0052】また、アラーム時は入力したオブジェクトがブリンク表示される様に構成したので、容易にオブジェクトの重複使用又未定義での使用を避けることができる。

【0053】さらに、画面に表示されたオブジェクトにカーソルを位置づけるとそのオブジェクトがオブジェクト一覧として表示される様に構成したので、オブジェクトを一元管理することができる。

【0054】また、画面に表示されたオブジェクトの名前を変更したときはオブジェクト属性テーブル上の属性データも変更される様に構成したので、容易にオブジェクトを一元管理することができる。

【0055】さらにまた、オブジェクト一覧のオブジェクトの定義日付にカーソルを位置づけると画面に表示されたオブジェクトが反転表示される様に構成したので、ソースコード中の定義箇所を容易に検索することができる。

【0056】また、オブジェクト一覧のオブジェクトの

12

参照日付にカーソルを位置づけると画面に表示されたオブジェクトが反転表示される様に構成したので、ソースコード中の参照箇所を容易に検索することができる。

【0057】さらに、オブジェクト一覧のオブジェクトの名称にカーソルを位置づけると画面に表示された定義文内にあるオブジェクトと参照文内にあるオブジェクトとがそれぞれ他のオブジェクトとは識別可能に表示される様に構成したので、ソースコード中の定義箇所及び参照箇所を容易に検索することができる。

【0058】また、ソースコード上の先頭に近いオブジェクトから順次オブジェクトが画面表示される様に構成したので、ソースコード中の定義箇所及び参照箇所を容易に検索することができる。

【図面の簡単な説明】

【図1】 この発明の実施の形態1のプログラミング支援装置を示す全体の構成図である。

【図2】 この発明の実施の形態1のオブジェクト編集装置の入力画面を示す構成図である。

【図3】 この発明の実施の形態1のオブジェクト属性テーブルを示す構成図である。

【図4】 この発明の実施の形態1のオブジェクト状態テーブルを示す構成図である。

【図5】 この発明の実施の形態1のソースコード編集装置（エディタ）の入力画面を示す構成図である。

【図6】 この発明の実施の形態1のソースコード情報を示す模式図である。

【図7】 この発明の実施の形態1のソースコード状態テーブルを示す構成図である。

【図8】 この発明の実施の形態1のソースコード編集装置（エディタ）主導の処理を示すフローチャートである。

【図9】 この発明の実施の形態1のオブジェクト追加処理を示すフローチャートである。

【図10】 この発明の実施の形態1のオブジェクト削除処理を示すフローチャートである。

【図11】 この発明の実施の形態1のオブジェクト表示処理を示すフローチャートである。

【図12】 この発明の実施の形態1のオブジェクト編集装置主導の処理を示すフローチャートである。

【図13】 この発明の実施の形態1のソースコード編集装置（エディタ）による変数定義の入力画面を示す構成図である。

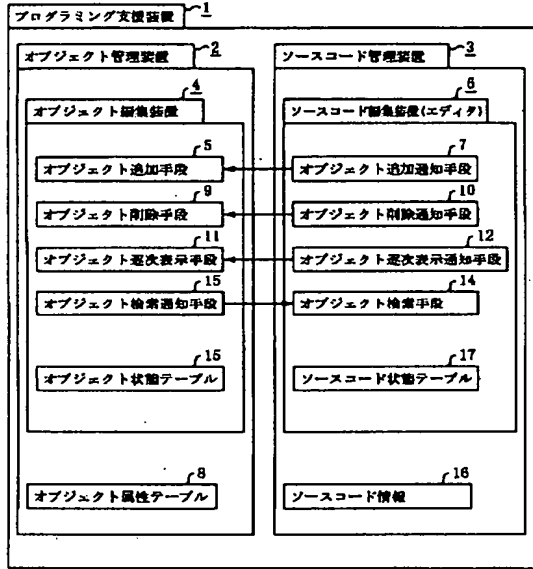
【図14】 この発明の実施の形態1のソースコード編集装置（エディタ）によるロケーション参照の表示画面を示す構成図である。

【図15】 この発明の実施の形態1のソースコード編集装置（エディタ）に関数表示の表示画面を示す構成図である。

【図16】 この発明の実施の形態1のオブジェクト編集装置のIO部品参照箇所検索コマンドの表示画面を示

す構成図である。

【図1】



【符号の説明】

【図2】

変数						
10部品						
関数						
タイプ						
表示順序						
○名前 ○タイプ ○長さ ○定義日付 ○参照日付 ○参照						
名前						
タイプ						
長さ						
定義日付						
参照日付						
コメント						
cBuf1	char	1	97/11/05 09:11:04	97/11/05 09:15:41	バッファ1	
cBuf2	char	2	97/11/05 09:11:12	97/11/05 09:15:54	バッファ2	
cBuf3	char	3	97/11/05 09:11:20	97/11/05 09:16:14	バッファ3	
cGroup1	char	6	97/11/05 09:18:20	97/11/05 09:19:15	グループ1	
cGroup2	char	6	97/11/05 09:18:36	97/11/05 09:19:21	グループ2	
cGroup3	char	6	97/11/05 09:18:43	97/11/05 09:19:42	グループ3	
cGroup4	char	6	97/11/05 09:18:50		グループ4	
cGroup5	char	6	97/11/05 09:18:59		グループ5	
cPerson1	char	8	97/11/02 14:13:16	97/11/02 14:15:02	人名1	
cPerson2	char	8	97/11/02 14:13:22	97/11/02 14:15:15	人名2	
cPerson3	char	8	97/11/02 14:13:32	97/11/02 14:15:37	人名3	
cPerson4	char	8	97/11/02 14:13:40	97/11/02 14:15:38	人名4	
cPerson5	char	8	97/11/02 14:13:52	97/11/02 15:12:56	人名5	
nLoop	int		97/11/05 20:11:00			
nLoopCo	int		97/11/05 20:12:24			
nList	int		97/10/24 08:10:37	97/11/25 12:59:25	リスト数	

【図4】

【図3】

オブジェクト属性テーブル									
変数									
名前	タイプ	長さ	定義日	参照日	コメント	定義ID/日付	参照ID/日付	参照箇所	参照箇所
.....							
10部品									
ファイル									
名前	定義日	参照日	コメント	参照ID/日付	参照箇所	参照箇所	参照箇所	参照箇所	参照箇所
.....							
関数									
名前	タイプ	長さ	定義日	参照日	コメント	定義ID/日付	参照ID/日付	参照箇所	参照箇所
.....							
引数									
名前	タイプ	長さ	定義日	参照日	コメント	定義ID/日付	参照ID/日付	参照箇所	参照箇所
.....							
ロケーション									
名前	定義日	参照日	コメント	定義ID/日付	参照箇所	参照箇所	参照箇所	参照箇所	参照箇所
.....							

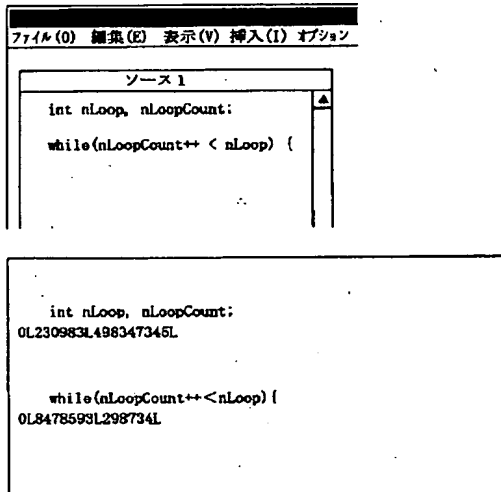
オブジェクト状態テーブル

オブジェクト名	バッファ
定義数	
参照数	
参照検索カウント	

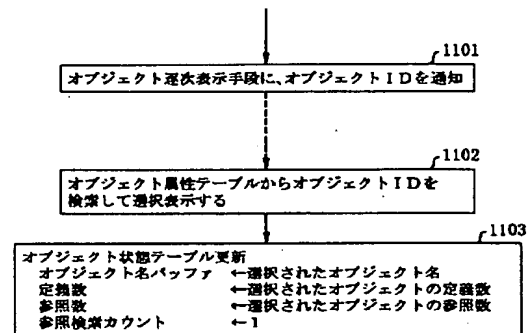
【図5】

ソース1		ソース2	
int nFunc10;		int nFunc20;	
int nPara1;			
int nPara2;			
{			
int nRtn;			
nRtn = nPara1 + nPara2;			
return nRtn;			
}			

【図6】

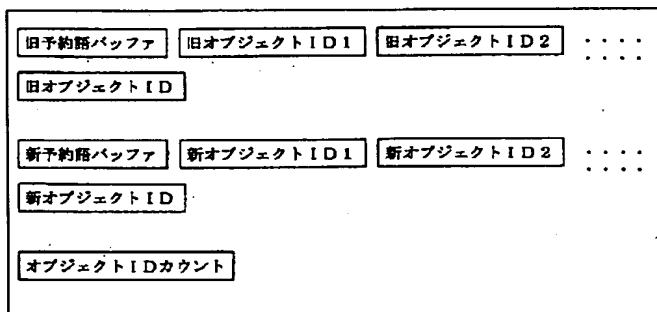


【図11】

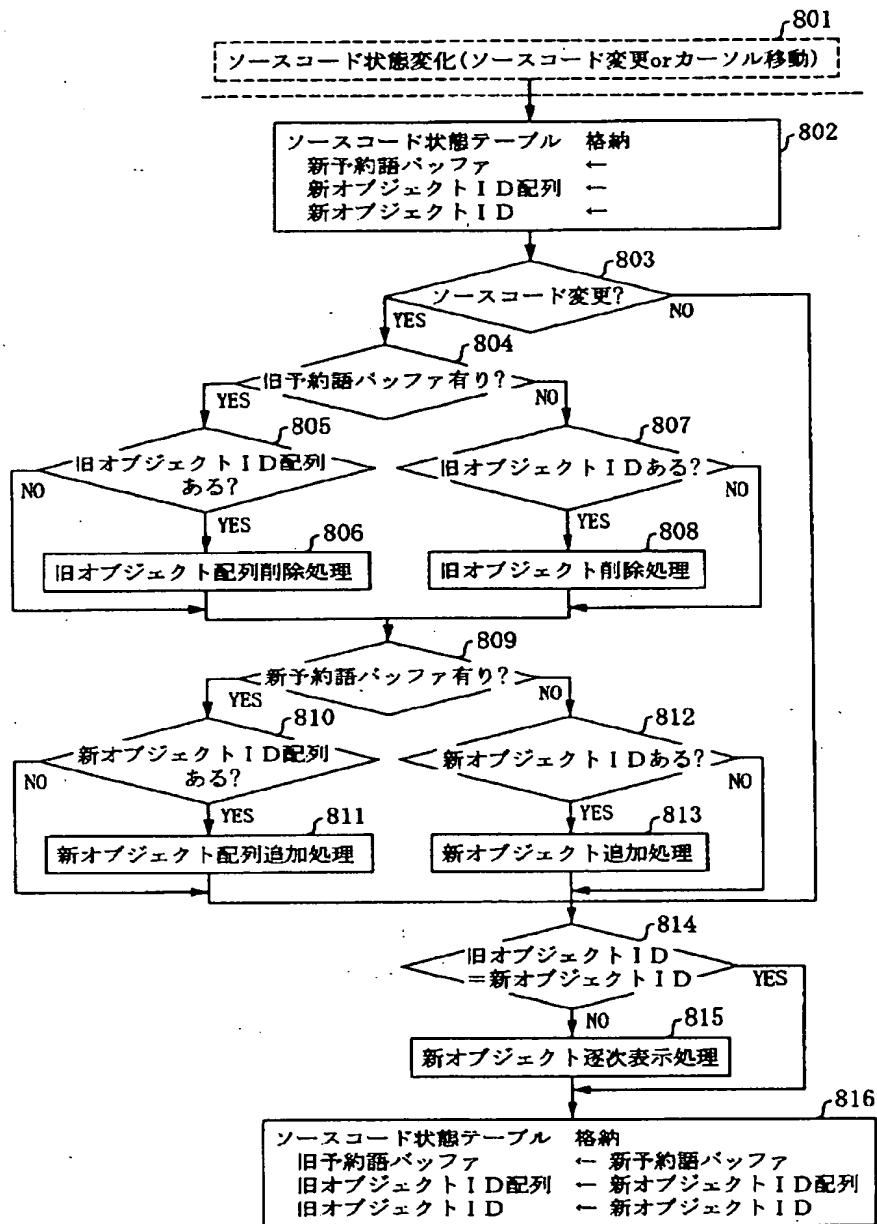


【図7】

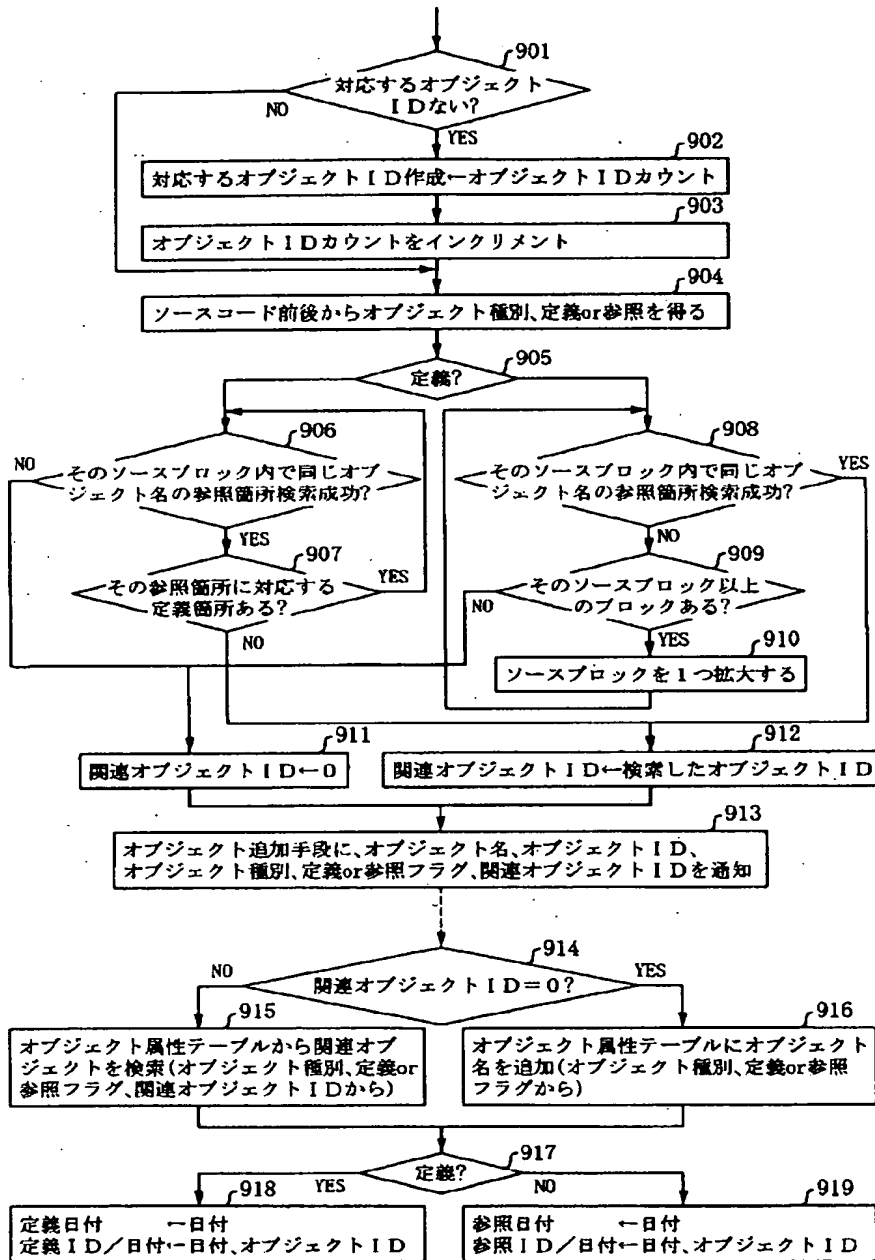
ソースコード状態テーブル



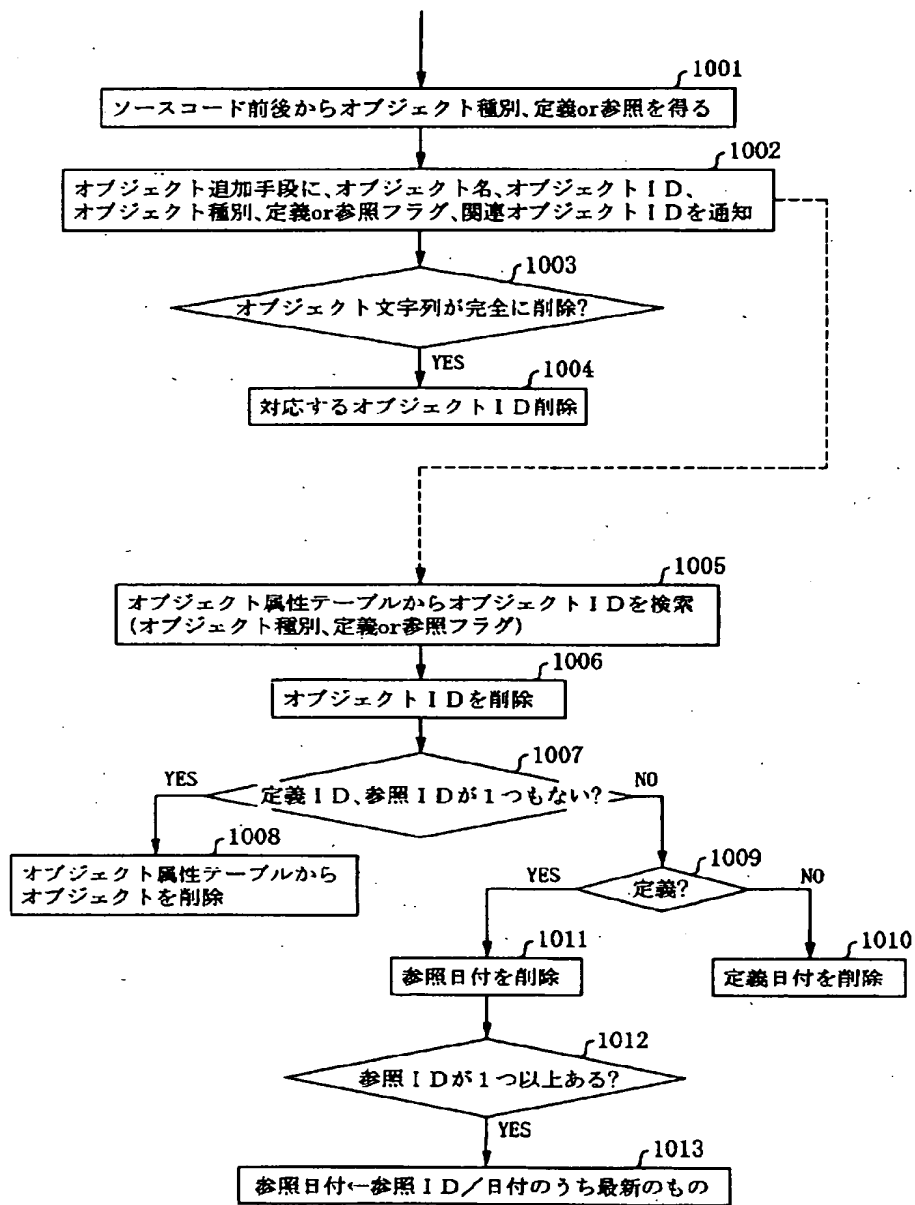
【図8】



【図9】



【図10】



```

graph TD
    1201[定義箇所検索コマンド] --> 1202[オブジェクトID←定義ID]
    1203{参照検索カウント=参照数?}
    1204[参照検索カウント-1]
    1205[参照検索カウントを1だけインクリメント]
    1206[オブジェクト検索手段に、オブジェクトIDを通知]
    1207[ソースコード情報からオブジェクトIDを検索して選択表示する]

    1202 --> 1203
    1203 -- YES --> 1204
    1203 -- NO --> 1205
    1204 --> 1206
    1205 --> 1206
    1206 --> 1207
  
```

【图 14】

```
int nLoop;
int nLoopCo;
```

```

ファイル(F) 編集(E) 表示(V) 挿入(I)

ソース 1
    if(nList > 0) goto NoError;
    else          goto Error1;
NoError:
    return 0;
Error1:
    return -1;
)

```

人数	100部品	回数	セッション
表示順序			
<input type="radio"/> 名前	<input type="radio"/> タイプ	<input type="radio"/> 長さ	<input type="radio"/> 定義日付
			<input type="radio"/> 参照日付
<input type="button" value="昇順"/>			

名前	タイプ	長さ	定義日付	参照日付	コメント
cBuf1	char	1	97/11/05 09:11:04	97/11/05 09:15:41	バッファ1
cBuf2	char	2	97/11/05 09:11:12	97/11/05 09:15:56	バッファ2
cBuf3	char	3	97/11/05 09:11:30	97/11/05 09:15:14	バッファ3
cGroup1	char	6	97/11/05 09:18:20	97/11/05 09:18:15	グループ1
cGroup2	char	6	97/11/05 09:18:36	97/11/05 09:19:21	グループ2
cGroup3	char	6	97/11/06 09:18:43	97/11/05 08:19:43	グループ3
cGroup4	char	6	97/11/06 09:18:50		グループ4
cGroup6	char	6	97/11/06 09:18:59		グループ6
cPerson1	char	8	97/11/02 14:13:12	97/11/02 14:15:02	人名1
cPerson2	char	8	97/11/02 14:13:25	97/11/02 14:16:15	人名2
cPerson3	char	8	97/11/02 14:13:32	97/11/02 14:15:27	人名3
cPerson4	char	8	97/11/02 14:13:40	97/11/02 14:16:38	人名4
cPerson5	char	8	97/11/02 14:13:52	97/11/05 15:18:56	人名5
nLoop	int		97/11/05 20:11:05		
nMain	int		97/11/05 20:12:24		
nList	int		97/10/24 08:10:37	97/11/25 12:59:25	リスト数

[illegible]

【図15】

ファイル(F) 編集(E) 表示(V) 挿入(I)

ソース 1

```

n2 = n1 * n1;
n3 = n1 + 10;
if (nComp | nf | (n2, n3) > 0) {
    n2 = n2 * 10;
}

return 0;

```

変数 | IO製品 | 関数 | コメント

表示順序
☐ 名前 ☐ タイプ ☐ 長さ ☐ 定義日付 ☐ 参照日付 昇順

関数

名前	タイプ	長さ	定義日付	参照日付	コメント
GetDate	void		97/10/29 13:50:11	97/10/30 14:31:29	日付を取得
DumpInf	void		97/11/01 09:32:53		情報をダンプ
Compare	int		97/11/01 15:45:13	97/11/02 20:12:45	整数を比較

定義箇所 参照箇所 情報転記

引数

名前	タイプ	長さ	定義日付	参照日付	コメント
nVal1	int		97/11/01 13:48:12		変数1
nVal2	int		97/11/01 13:45:10		変数2

定義箇所 参照箇所 情報転記

【図16】

変数 | IO製品 | 引数 | コメント

ファイル | 検索 | 通信データ

表示順序
☐ 名前 ☐ 定義日付 ☐ 参照日付 昇順

部品一覧

名前	定義日付	参照日付	コメント
Master	97/10/11 13:12:46	97/11/01 11:31:26	マスターファイル
Update1	97/10/13 11:23:56		更新ファイル1
Update2	97/10/25 13:32:08	97/10/30 14:02:13	更新ファイル2
Output	97/11/01 20:11:21		出力ファイル

参照箇所 情報転記

部品詳細

名前	タイプ	長さ	参照日付	コメント
Record1			97/10/02 08:12:24	レコード1(このレコードにおいて)
Code	char	4	97/10/02 08:11:12	製品コード
Date	char	8	97/10/02 08:11:23	日付
User1	char	8	97/10/02 08:11:45	ユーザ名

参照箇所 情報転記

ファイル(F) 編集(E) 表示(V) 挿入(I)

ソース 1

```

hFile = OpenFile("Master", ofFile,
_lread(hFile, cBuf9, 5);
_lolose(hFile);

```